Victorian 6502 User Group Newsletter

KAOS

For People Who Have Got Smart

Hardware: . .. . David Anear
Software:.. .. .. Jeff Rae
Languages: . .. . Tony Durrant
Amateur Radio: Rod Drysdale VK3BYU
Education:.. .. .. Noel Dollman
Library: .. .. Jannene
Tape Library: .. .. Ron Cork
Newsletter: .. .. Ian Eyles
Sym: .. .. Brian Campbell
Atari:.. .. ..Gerry McCaughey
Secretary: . .. . Rosemary Eyles

OSI SYM KIM AIM ATARI APPLE UK101 ORANGE

Registered by Australia Post
Publication No. VBG4212

ADDRESS ALL CORRESPONDENCE TO 10 FORBES ST., ESSENDON, VIC. 3040

Vol.2 No.5 February 1982

Please note that if you have not renewed your membership by the 14th March, this is the last newsletter you will receive, so if you plan to remain a member of KAOS, make haste and save us the bother of taking your name of the mailing list and then having to put it back on again.

Because of a mix-up with registration of the name The Australian Source (TAS) is now called - wait for it - The Australian Beginning (TAB?????)..... If you're not careful with your modem you may find yourself backing the winner of the Cup. We had hoped to arrange a demonstration at the February meeting but because of the problems of getting a large screen that everyone could clearly see, the demonstration has been postponed to a later meeting.

If you are not using your computer and would like to sell it, write to us, stating all the details and we will advertise it (for free) in the newsletter. If you have sold your computer, we would appreciate you dropping us a line to tell us who the new owner is, so that we can contact him or her and tell them about KAOS.

We had a phone call from David Tasker this week and his new business is now running smoothly which means he has more free time to work on the new video board, which is nearly completed. If there are no complications we will be able to let you know when it will be available in the March newsletter. The GTBUG is finished except for documentation, we hope to pin Tony down and get this done when he comes back from his holiday..

The next meeting will be held on Sunday 28th February at 2pm at the Essendon Primary School, corner of Raleigh St and Nicholson St, Essendon. Would the usual early arrivers please note that the children from the school will be in early for their lesson.

M.A.C.E.

Hello again.

What's new for the fantastic ATARI this month? At last some hardware is starting to surface. First there is a new I/O board that fits into the third memory slot on the 800. The main advantage is the ability to use a direct parallel interface instead of the present method of going out through the serial port, an interface, then converting back to serial. If this could be used to connect a disk drive, it would speed it up by a factor of 8 to 10.

They are also going to have an 8" disk ready soon. More info from MICROLINK 2078C Walsh Ave., Santa Clara, California.

If you have 4 to 6 thousand dollars to spare, you can buy a Winchester Disk System. They are from CORVUS Systems Inc., each unit comes with an intelligent controller which provides high speed operation that requires virtually no software support and are made in 5 to 20 megabyte versions.

ATARI Pascal will be released this winter, it will be sold through the Atari Program Exchange for about $50.00, compare that to the Apple. It will compile to 6502 code rather than P code. This gives the ATARI a very large language library, Basic, Basic A+, Forth, Lisp, Pilot, WSFN, various assemblers and shortly Microsoft Basic, various macro assemblers, Tiny C and Logo. Quite a collection and even more remarkable is the extremely low cost compared to any other computer.

New from Atari is the Book Keeper, it's a disk based program and comes with a numeric keypad, the Atari Pacman (at last) and Centipede. John Bell from Crystalware, has started a new company called U.F.O. SOFTWARE. He will market 16K cartridges, he already markets the best third party software for the Atari. A new cartridge company is K-BYTE. The first program is called K-RAZY SHOOTOUT, it is a superb game, it's a duplicate of BERZERK for the 400/800. They also plan on releasing K-DOS, also K-STAR PATROL, a defender type game and last of all a invaders type game. All of these are excellent, better than most of the Atari cartridges. Contact K-BYTE, P.O. BOX 456, 1755 AUSTIN, TROY, M148099, tell them you heard from MACE.

ATARI have just sent me an advanced Programmers guide for the ATARI. It is very well written and very helpful. See me if you want to borrow it for copying, it is about 200 pages so have plenty of money for copying.

ATARI RUMOURS:- It seems Atari are are working on a voice recognition unit,they are very advanced so don't expect to wait to long for it.

$$

FOR SALE

1	Teleray	V.D.U.	$200.00	obo
	RS 232	110 - 4800 Baud		
	80 X 25			
1	ASR 33 Teletype		$300.00	obo
4	8K Tasker Buss	RAM Boards	$ 60.00	each
			$200.00	the lot

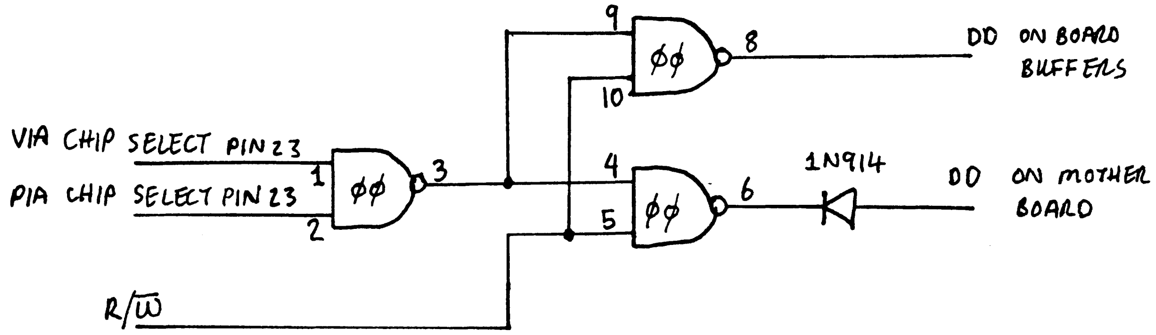
Warren Schaeche
ph.

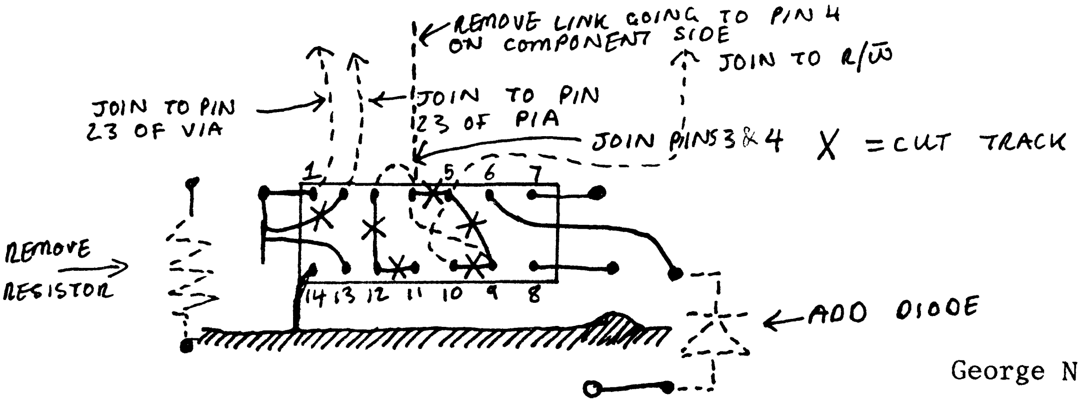
THE VIA/PIA BOARD FIX

Those of you who are using disk on the Tasker Buss system will have found that your I/O board will not work correctly.

This is because David Tasker derived his DD (data direction) signal, which controls the direction of flow on the data Buss, by using the C0/C8 decoded line, which means that when any byte within that 2K region is addressed, the I/O board was always functioning.

A simple solution to this problem can be made by changing the logic of the DD line. Unfortunately, the 7412 chip previously used is inadequate so we have to use 7400 in its place. The existing socket and tracks of the 7412 can be altered to allow the 7400 to plug back in its place. This is shown below.





George Nikolaidis

****

MEDIUM RESOLUTION GRAPHICS

For those of you who have enquired on the Medium resolution graphics for the Series I machines you will be happy to know that the mod is very easy. The documentation for the mod is almost complete with minor changes to be made. The machine code graphics routines have been converted but await documentation.

The only problem I can see with this mod is that it uses the RTS line to select between the two character sets. Whilst this is not a problem in itself, the problem is that you may wish to use the RTS line at some time in the future. This will be true if you wish to use some printers or a modem. To also have this mod would mean manually selecting with a switch which option you would wish to run.

If you are still interested in the mod then see me at the next meeting or contact me on

Kelvin Eldridge

MODIFICATION TO MAKE C4 BOARD INTO C1

If, after building the C1-C4 video board, you are still keen for punishment, then this is for you. You can modify the C4 board so that you are able to display graphics in the C1 format ie. adding 32 to the screen address to move down one one.

PARTS NEEDED

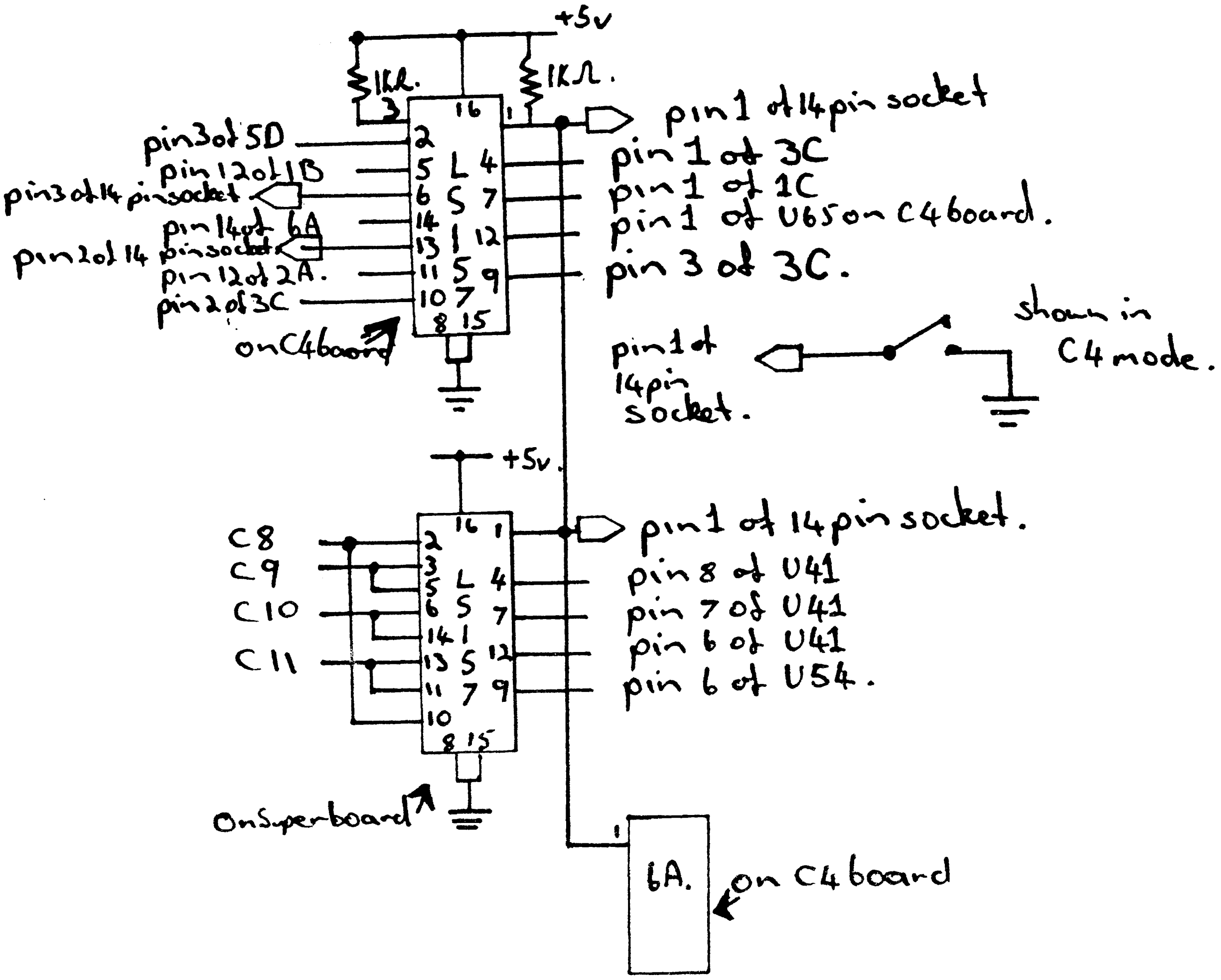
2 X 74LS157
2 X 16 pin sockets
2 X 14 pin sockets
2 X 14 header plug (solder type)
Ribbon cable to suit above and some jumper wire ie. wire-wrap wire
Install a 14 and a 16 pin socket on the Superboard
Remove the jumpers installed in steps 14 and 15 of the C4 instructions
Connect pin 6 of U41 to pin 12 of the 16 pin socket
Connect pin 7 of U41 to pin 7 of the 16 pin socket
Connect pin 6 of U41 to pin 4 of the 16 pin socket
Place jumpers between the following pins of the 16 pin socket, pin 3 & pin 5; pin 6 & pin 14; pin 13 & pin 11; pin 2 & pin 10.
Connect +5 volts to pin 16 and ground to pins 8 and 15.
Remove the jumper from between pin 6 of U54 and pin 14 of U60
Connect pin 6 U54 to pin 9 of the 16 pin socket
Connect pin 11 U60 to pin 11 of the 16 pin socket
Connect pin 11 U59 to pin 3 of 14 pin socket
Connect pin 11 U61 to pin 2 of the 14 pin socket

On the C1-C4 board install the 16 pin and 14 pin sockets in spaces provided.
Connect pin 15 and pin 8 to ground and pin 16 to +5 volts
Connect a 1K resistor between +5 volts and pin 3
Connect a resistor between +5 volts and pin 1 of 16 pin socket
Using a sharp knife cut the copper trace between pin 1 of 6A and +5 volts
Connect a jumper between pin 1 of 16 pin socket and pin 1 of 6A and pin 1 of the 14 pin socket
Next comes a bit of desoldering to rèmove some links between chips. Follow the copper trace from the point stated and remove the first link that passes from one side to the other.
Remove link from pin 1 U65 on the C4 board.
Remove link from pin 1 of 3C.
Remove the link from pin 3 of 3C
Remove the link from pin 12 of socket 1B
Connect pin 4 of the 16 pin socket to pin 1 of 3C
Connect pin 2 of the 16 pin socket to pin 3 of 5D
Connect pin 5 of 16 pin socket to pin 12 of socket 1B
Connect pin 7 of 16 pin socket to pin 1 of 1C
Connect pin 14 of 16 pin socket to pin 14 of 6A
Connect pin 12 of 16 pin socket to pin 1 of U65 on the C4 board
Connect pin 11 of 16 pin socket to pin 12 of 2A
Connect pin 10 of 16 pin socket to pin 2 of 3C
Connect pin 9 of 16 pin socket to pin 3 of 3C
Connect pin 13 of 16 pin socket to pin 2 of the 14 pin socket
Connect pin 6 of 16 pin socket to pin 3 of 14 pin socket
Connect pin 16 of 14 pin socket to ground

Now solder directly to the bottom of the 14 pin socket a 14 strand ribbon cable. Care should be taken to note the way the header plug connects to this cable so as to get the right way the first time. Solder the other end of the cable to plug. Now solder 2 wires to the last header plug and to a switch as per circuit, and insert this into the 14 pin socket from the top. Mount the swich on the front or back panel, this switch will change the display format.

Last but not least, re-assemble the Superboard and the C4 board and apply power. It should wake up and you should be able to operate the switch and swap between the two formats. If not, happy hunting!!!!!!!!

As a last resort my phone number is   Jeff Rae.



Modification to Make C4 board into C1

FOR SALE

TRENDCOM 200   thermal printer $380.00 ono
(80 characters wide)
18 months old   excellent condition

Mark Blakey
ph.

FOR SALE

C1P Series II $500.00
8K RAM   DABUG III
RS232 port   sound port
2 graphics modes polled keyboard
includes all manuals   Game cassettes extra!

call   10am - 1pm   or   6pm - 11pm
ask for Nick

EPROM BOARD MODIFICATION

A lot of you in the past year have contructed the Tasker Buss system and might have purchased an EPROM board which you are not currently using. Lately, a new static RAM chip has become readily available at a reasonable cost which is pin compatable with 2716 EPROMs. With slight modifications, your un-utilized EPROM board may be converted to an 8K RAM board. Some of these chips, such as the Hitachi 6116 made using C MOS technology, which means that their power consumption is so low that the contents can be maintained for a very long period using battery back-up.

NOTE: READ ALL INSTRUCTIONS BEFORE STARTING PLEASE

FOLLOW THESE STEP BY STEP INSTRUCTIONS

STEP 1

The current design of the EPROM board has a data buffer which only allows for the data to flow in one direction. Unfortunately this is not suitable for a RAM board which requires the data to flow in both directions (bi-directional). For simplicity we remove the 74LS241 buffer and by-pass it. This is achieved by joining the following pins together.
2 to 18   4 to 16   6 to 14   8 to 12   3 to 17
5 to 15   7 to 13   9 to 11
This is shown on the diagram below, section A

STEP 2

Since RAMs require a R/W line we have to provide one by taking it off the Buss and fixing it to pin 21 of each chip. Unfortunately pin 21 is tied to plus 5 volts and has to be cut as shown in section B.

STEP 3

We then place a link from the R/W track to pin 21 as shown in section C.

STEP 4

The current circuit is designed so pin 20 on the EPROM chips acts as a chip select. The new RAM chips require the chip select on pin 18, so each chip select has to be moved from pin 20 to pin 18. This is achieved by removing the links joined to pin 20, running parallel with the EPROM sockets on the component side of the board, then rejoining them to pin 18 on each chip, see section D.

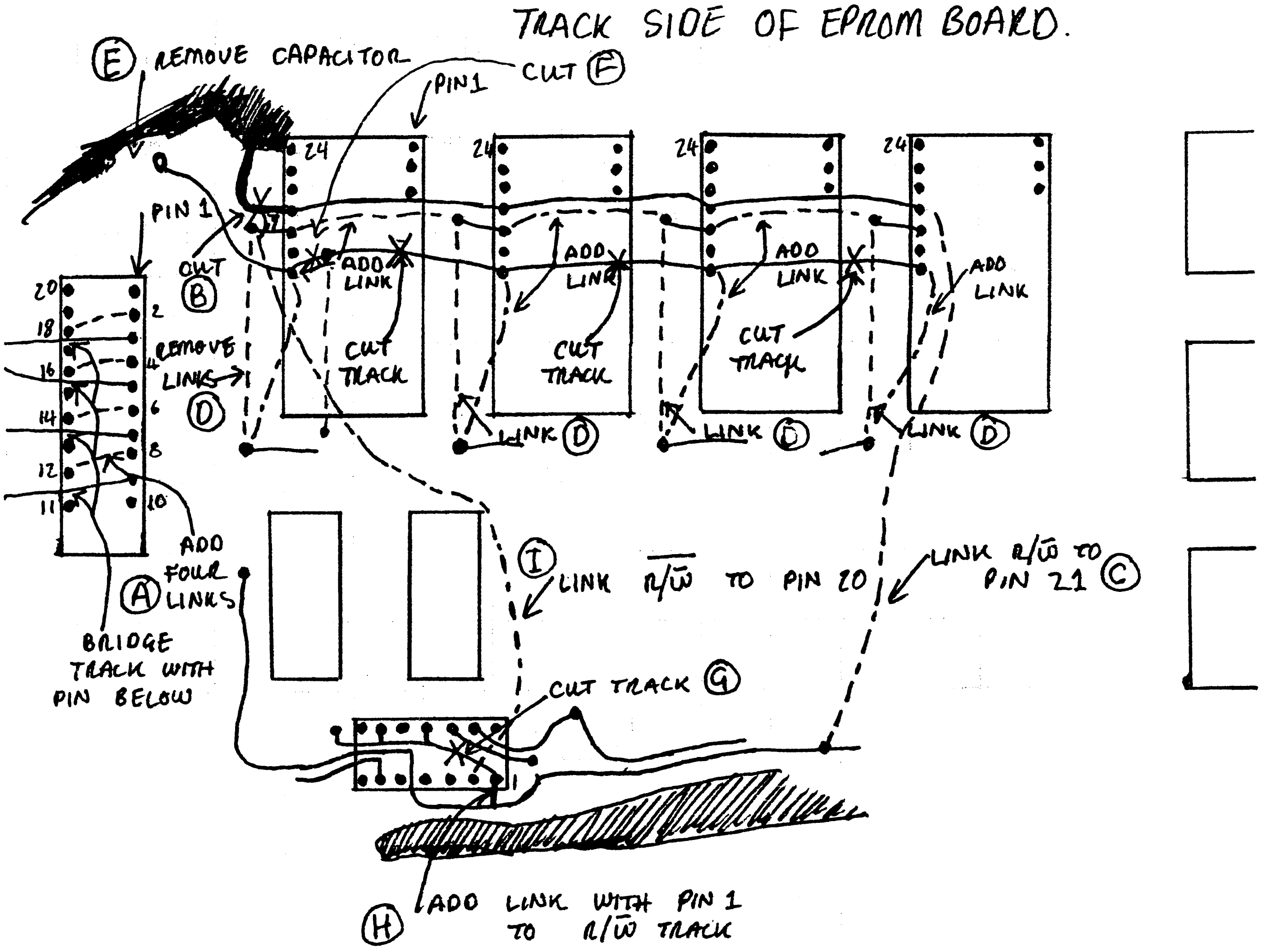
STEP 5

Remove the capacitor as shown in section E and cut the tracks running between all the pin 18s and isolate the link joined to pin 18 on the chip closest to the buffer, see section F.

STEP 6

The timing diagram of pin 20 on the new RAM chips, specify that the levels work logically opposite to the R/W line. There is no provision on the EPROM board for a NOT R/W signal, therefore we have to generate one using one of the spare inverter gates on the board. The most appropriate one is the one on pin 1 and 2 of the 74LS04. Cut the track going to pin 1 from pin 11 (see section G) and then link pin 1 to the track below it which happens to be the R/W line. See section H. We then link pin 2 to pin 20 on each chip as shown in section I. Link all the pin 18s of each chip together.

George Nikolaidis



****

SYM-POSIUM

SUPERMON - THE CASSETTE INTERFACE

SUPERMON is the name of the powerful 4K monitor program in the SYM. After releasing the first version (V1.0), it was found that the 1500 baud high speed cassette interface was not very reliable. A second version of the monitor program (V1.1) was written to correct this problem.

Because the data is being sent out at a fairly high rate, you may have some difficulty getting the interface up and running. If you are having trouble, here are a few tips :-

1. Cheap cassette recorders and tapes are out! The $35 recorders do not have a frequency response or tape speed regulator capable of replaying (or recording, for that matter) with enough accuracy for a 1500 baud data channel. There is no need to spend thousands of dollars on a top line cassette deck, but something of a reasonable quality will do wonders. I use an Audio Reflex DC-26 cassette deck which cost me $85.00 and when combined with the $1.00 tapes available through KAOS, I have a 100% reliable system.

2. If you have SUPERMON V1.0 then get V1.1. Even if your cassette system is working well, get the other monitor anyway as it has many other enhancements which you will see later on.

PTO

The easiest way to tell which version of the monitor you have is to press the CR button on the hex keypad after resetting your SYM. If you have the first version of the monitor, the display will show 'SY1.0 ..' while the second version will show 'SY1.1 ..'.

In V1.1, many constants in the cassette routines were moved into system RAM. Location $A630 controls the length of the sync leader. It currently has the value $04, but dropping to $01 will reduce the leader to a minimum. Setting it to $FF is the best way to make a sync tape.

The best trick for V1.1 is to fiddle the values in locations $A632, $A635 and $A63C. These locations control the major timing loops. The cassette interface speed is software controlled, so by halving the values in these three locations your cassette interface will operate at 3000 baud with absolutely no hardware changes. If your tape deck is good enough, try halving it again. You may be lucky enough to have a reliable 6000 baud cassette system. Who needs a disk?

When writing programs which use subroutines in the cassette interface part of the monitor, it is important to note which version you are using as their start locations are different in each monitor. This does not effect other SUPERMON subroutines, but more about this next month.

NEXT MONTH:- SUPERMON ROUTINES

SUPERMON consists mainly of subroutines which can be very useful in user programs. SUPERMON itself can be used as a subroutine. Next month we will look at ways to use the subroutines and list a few that are not mentioned in the book.

Brian Campbell

****

SCREEN FREEZE

This program was born from the fact that while modifying some programs on the C4, I was annoyed by the problem that if you ran a program that swapped the screen format, you had to type in POKE 56832,1 before you could read all that was being displayed on the screen. With this program hooked into the CTRL C you can freeze the screen at any time whether you are printing or poking characters to the screen. By depressing CTRL and S at the same time the screen will freeze and stay frozen until another key is pressed. Also by using this same hook you can cause the C4 to switch to 64 screen when you press CTRL C. This is handy when you are developing or modifying a program as you can then list with a 64 screen

There are two versions of this program, one each for the C4 and C1. To use this program do a cold start, then use the monitor to enter the code in the third column into the machine. Once you have entered the code, modify locations #0000 - #0002 to read 4C 4B 02 for the C4 and 4C 3C 02 for the C1, then do a warm start and the added control keys should now work. If it does not work, check the code you entered against the original.

One final note, line 210 in the source code turns on DABUG so if you have a standard monitor, fill the relevant location with NOP's (EA).

Jeff Rae

SOURCE FOR C4

```
10  0222    A901              LDA #$01
20  0224    8D00DF            STA $DF00
30  0227    2C00DF            BIT $DF00
40  022A    501C              BVC END
50  022C    A908              LDA #$08
60  022E    8D00DF            STA $DF00
70  0231    2C00DF            BIT $DF00
80  0234    1003              BPL C
90  0236    2000FD            JSR $FD00
100 0239    A904     C        LDA #$04     ;Routine to swap screen size
110 023B    8D00DF            STA $DF00    ; when CTRL C is hit (C4 only)
120 023E    2C00DF            BIT $DF00    ;        "          "
130 0241    5005              BVC END      ;        "          "
140 0243    A901              LDA #$01     ;        "          "
150 0245    8D00DE            STA $DE00    ;        "          "
160 0248    4CB9FF   END      JMP $FFB9
170 024B    A922              LDA #$22
180 024D    8D1C02            STA $021C
190 0250    A902              LDA #$02
200 0252    8D1D02            STA $021D
210 0255    20E2F9            JSR $F9E2
220 0258    4C74A2            JMP $A274
```

SOURCE FOR C1

```
10  0222    A9FE              LDA #$FE
20  0224    8D00DF            STA $DF00
30  0227    2C00DF            BIT $DF00
40  022A    700D              BVS END
50  022C    A9F7              LDA #$F7
60  022E    8D00DF            STA $DF00
70  0231    2C00DF            BIT $DF00
80  0234    3003              BMI END
90  0236    2000FD            JSR $FD00

160 0239    4CB9FF   END      JMP $FFB9
170 023C    A922              LDA #$22
180 023E    8D1C02            STA $021C
190 0241    A902              LDA #$02
200 0243    8D1D02            STA $021D
210 0246    201AFA            JSR $FA1A
220 0249    4C74A2            JMP $A274
```

****

TELETYPE

To use the Teletype routine with DABUG III change the first instruction from JSR $FF69 to JSR $FB00

EXTENDED MONITOR

If you use the Extended Monitor to print out disassembled code change $099D from 17 to 46 to print one A4 page before stopping.

John Whitehead

****

MACHINE CODE PROGRAMMING

The main use of the X and Y registers is as counters to be used for indexed addressing and delay loops.

When writing M/Code programs, especially real time games, it is often necessary to include delay loops to slow the action down.

Load the programs below in the usual way, then Break and either cold or warm start to get into Basic.
Enter the following in immediate mode:

```
POKE11,34:POKE12,2:X=USR(X)
```

This set up the USR(X) vectors and then runs the M/Code program, the RTS at the end of the programs will return the machine to Basic.

```
10 0222  A2FF       LDX #$FF       ;Initialize X to required delay
20 0224  CA   LOOP  DEX
30 0225  DOFD       BNE LOOP
40 0227  60         RTS            ;Return to main program
```

Note that the count is Decremented from (required delay) to 0. In the X and Y registers the Z flag is set when the count reaches 0, this means that the end of the count can be detected by a Branch instruction. If the count was from 0 to a value and used Increment, then a Compare instruction would be required.

In most cases the value for the delay will be found by trial and error but it can be calculated by using a 6502 instruction set chart. In the example above LDX takes 2 cycles, DEX 2 cycles and BNE 3 cycles, therefore the delay will be 2 + (5 * 255) cycles, at a clock frequency of 1 MHz this = 1277 / 1MHz = 0.001277 seconds. To obtain a longer delay we need to add a second loop:

```
10 0222  AOFF          LDY #$FF     ;Initialize Y register
20 0224  A2FF   LOOP1  LDX #$FF     ;     "     X    "
30 0226  CA     LOOP2  DEX
40 0227  DOFD          BNE LOOP2
50 0229  88            DEY
60 022A  DOF8          BNE LOOP1
70 022C  60            RTS          ;Return to main program
```

In this routine X will count down to 0, then Y will Decrement by one and X will be reset to the original value. This will be repeated until Y=0. To calculate the delay for The delay for this routine is: (2+2+(5*X)+5+2-1)*Y = (2+2+(5*255)+5+2-1)*255 = 327675 = 327675/1MHz = 0.327 seconds. In the section of the equation underlined the +2 is to re-initialize the X register and the -1 is because the BNE only takes 2 cycles when it does not loop.

The next program stores the start address of the screen RAM in $00-$01 and adds the contents of the Y register to it to get the next location (line 280). At the end of each line (line 300) 32 is added to the address in $00-$01 to move to the next line down.

The 8 graphics characters 80 to 87 Hex are used in sequence to sweep a narrow band down the screen, and a 20 Hex is used after each 8 passes to clear the screen.

The delay routine at line 330- 380 alters the speed and if the code from $023E to $0242 is replaced with NOP's (EA) a row of bars will be drawn down the screen

If you have a C4 use the numbers in brackets.

```
 10 0222              *=$0222
 20 0222  A9D0                LDA #$D0          ;Load screen address and
 30 0224  8501                STA $1            ; save at $00 and $01
 40 0226  A900                LDA #$0
 50 0228  8500                STA $0
 60 022A  A987        LOOP4   LDA #$87          ;Load graphics character
 70 022C  8502                STA $02           ;Store at $02
 80 022E  A502        LOOP1   LDA $2
 90 0230  205502              JSR DSPLAY
100 0233  205F02              JSR WAIT
110 0236  C602                DEC $2            ;Next character
120 0238  A502                LDA $2
130 023A  C980                CMP #$80          ;Last char required
140 023C  B0F0                BCS LOOP1         ;No, loop
150 023E  A920                LDA #$20
160 0240  205502              JSR DSPLAY
170 0243  18                  CLC
180 0244  A500                LDA $0            ;Add 32 to address in $00
190 0245  6920 (40)           ADC #$20(40)      ; to move down one line
200 0248  8500                STA $0
210 024A  A501                LDA $1
220 024C  6900                ADC #$0           ;Add 0 PLUS ANY CARRY FROM
230 024E  8501                STA $1            ; FIRST ADC to $01
240 0250  C9D4 (D7)           CMP #$D4 (D7)     ;Is it bottom of screen
250 0252  90D6                BCC LOOP4         ;No, do next line
260 0254  60                  RTS
270 0255  A000        DSPLAY  LDY #$00
280 0257  9100        LOOP2   STA ($0),Y
290 0259  C8                  INY
300 025A  C020 (40)           CPY #$20 (40)     ;End of line?
310 025C  90F9                BCC LOOP2
320 025E  60                  RTS
330 025F  A001        WAIT    LDY #$01
340 0261  A201        LOOP5   LDX #$01
350 0263  CA          LOOP3   DEX
360 0264  D0FD                BNE LOOP3
370 0266  88                  DEY
380 0267  D0F8                BNE LOOP5
390 0269  60                  RTS
```

As an exercise you could try writing a program to make the bars travel up the screen.

The next program uses some of the same routines, but instead of drawing lines it looks for the letter 'A' in the screen RAM, if one is found it is changed to a CHR BB and the program scan the rest of the screen, then returns to the main program.

This type of program could be used in games to move characters around the screen, for instance, if 'Y' was incremented before the BEQ to CHANGE all the 'A's on the screen would be moved one place to the right

To run the program, load it with the monitor, then cold start and write a Basic program to put some text on the screen eg.

```
10 FORX=1TO10;INPUT A$(X):NEXT
20 POKE 11,34:POKE 12,2:X=USR(X)
```

Enter the 10 lines of text and when return is pressed after the last input, all the 'A's will be changed. The character to be changed is at $022F and the character to be inserted is at $024C.

The numbers in brackets are for the C4 machine.

```
 10 0222                  *=$0222
 20 0222   A9D0                     LDA #$D0           ;Load screen address and
 30 0224   8501                     STA $1             ; save at $00 and $01
 40 0226   A900                     LDA #$0
 50 0228   8500                     STA $0
 60 022A   A000    RESET            LDY #$00           ;Set counter to zero
 70 022C   B100    LOOP             LDA ($0),Y         ;LDA from address in
 75                                                    ;$00-$01 plus Y
 80 022E   C941                     CMP #$41           ;Is it 'A'
 90 0230   F019                     BEQ CHANGE         ;Yes, go to CHANGE routine
100 0232   C8      COUNT            INY
110 0233   C020 (40)                CPY #$20 (40)      ;Is count 32 (line length)
120 0235   DoF5                     BNE LOOP           ;No, look at next address
130 0237   18                       CLC
140 0238   A500                     LDA $0
150 023A   6920 (40)                ADC #$20 (40)      ;Add 32 to address in $00
160 023C   8500                     STA $0             ; to move down one line
170 023E   A501                     LDA $1
180 0240   6900                     ADC #$0            ;Add 0 PLUS ANY CARRY FROM
190 0242   8501                     STA $1             ; FIRST ADC to $01
195 0244   A501                     LDA $1
200 0246   C9D4 (D7)                CMP #$D4 (D7)      ;Is it bottom of screen
210 0248   D0E0                     BNE RESET          ;No, do next line
220 024A   60                       RTS
230 024B   A9BB    CHANGE           LDA #$BB           ;Load new character and
240 024D   9100                     STA ($0),Y         ; store at current address
250 024F   4C3202                   JMP COUNT
```

****

OOOPS!!!!!

Several mistakes were made when the Competion Worm program was re-typed for the news letter.

```
Line 50   Last colon should be semi-colon
 "   810  BP=(C2+WI*RND(8)-LL*INT(23*RND(8)))
 "   815  IFPEEK(BP)   32THEN810
 "   950  REM CONTROL KEYS SELECTED TO SUIT JOYSTICKS
 "   960  RETURN
```